17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

SpaceOps-2023, ID # 150

AI Deep Learning Radar Target Detection & Classification Model for Real Time Space Debris

AlMalki J. Muneera[a]*, Senan A. Reem[b]

[a] *Department of Bahrain Space Team, National Space Science Agency, Road 1510, Block, 115, Building 702 P.O. Box 51115, Hidd, Kingdom of Bahrain,* muneera.almalki@nssa.gov.bh
[b] *Department of Bahrain Space Team, National Space Science Agency, Road 1510, Block, 115, Building 702 P.O. Box 51115, Hidd, Kingdom of Bahrain,* reem.senan@nssa.gov.bh

* Corresponding Author

Abstract

Space debris has been a subject of concern for all, as it sets a major threat to space activities. This threat has propelled various research in the field of detecting, monitoring, predicting, and reducing space debris and their trajectories using numerous techniques and methods. The detection of space debris is crucial to aid in collision avoidance which in the event of collision, it may harm spacecrafts that are orbiting, especially with the continuous increase of the debris. This paper investigates the detection and classification of the space debris using radar detection data for an optimized low complexity and low-cost system. An Artificial Intelligence (AI), deep learning (DL) model was developed using Deep Neural Network (DNN) for target detection and classification of real-time space debris. The system was shown to distinguish and classify objects adequately, since, the model followed an architecture of target list, classification, data labelling, and filtering. Subsequently, an analysis of the DL model was conducted for an approach of clustering in the place of classification. The model developed can be integrated in numerous payload sensors and other radar gadgets that will aid in space debris monitoring, collision avoidance, and decision makers in the long-term sustainability of outer space activities.
Keywords: Space Debris, Artificial Intelligence, Deep Learning, Deep Neural Network, Radar Detection, Classification.

Acronyms/Abbreviations

4D	4 Dimensional
BNB	Bernoulli Naive Bayes
DNN	Deep Neural Networks
FOV	Field of View
GNB	Gaussian Naive Bayes
GPS	Global Positioning System
KNN	K-Nearest Neighbour
LEO	Low Earth Orbit
ML	Machine Learning
NASA	National Aeronautics and Space Administration
RADAR	National Aeronautics and Space Administration
RCS	Radar Cross Section
RFC	Random Forest Classifier
SDG	Sustainable Development Goals
SVM	Support Vector Machine

1. Introduction

Globally, it is recognized that space debris is a growing issue that makes access to Earth's orbits progressively more difficult, as 99% of operational risks of spacecrafts come from space debris and other spacecraft collisions [1]. Space debris as defined by the National Aeronautics and Space Administration (NASA) is any artificial object around Earth's orbit that is no longer functional and useful [2]. Space debris entails high costs due to space missions suffering losses through collision or impact which will negatively affect space applications such as Earth observation and weather forecasting, as well as the costs of collision avoidance, debris mitigation, and removal [3]. The number of launched satellites has tremendously increased to more than a thousand per year in 2021, whereas the number of re-entries has remained the same, a few hundred per year, thus making the number of space objects and debris continuously increase to more than 140 million fragments [4]. This increasing number of debris is a constant threat to operating spacecrafts. This is an alarming issue, especially for spacefaring countries, as it is considered the number one risk in space missions, as well as their cascading effect on achieving the sustainable development goals (SDGs), as 40% of the 17 SDGs, cannot be achieved without space utilization [5]. Space debris from sizes of 1 mm to 10 cm needs enhanced tracking and monitoring methods to effectively monitor and track space objects [4]. Utilizing specialized space-based sensors/radars is one option as a solution to monitoring the debris, as an orbital detector would have a number of advantages over ground-based methods, including independence from weather conditions and vast coverage [6].

The detection of space debris has focused on 10 cm and above sizes, thus research on detecting smaller space debris is still considered ongoing, and the detection performance on small debris has continued to be low [7]. A Radio Detection and Ranging (RADAR) sensor offers long-range detection suitable for space that can be utilized to detect small debris. It is small in size, inexpensive, and has low power consumption with the ability to detect objects and provide spatial information with accurate doppler velocity, hence it is suitable to be used in small satellites such as CubeSats for space debris detection [8]. Using Deep Neural Networks (DNN) has enhanced the RADAR's ability to provide the classification and shape of the detected objects [8]. DNN has recently become a powerful emerging machine-learning (ML) model that is used heavily for object detection and classification [9,10].

The research in the field of space debris detection and classification has been of interest as of late, due to the increased threat to present and future space missions. In 2015, a study by A. Morselli et al. [11] developed a high-sensitivity radar sensor for detecting space debris. It has enabled the discerning of the trajectory of the space debris object in relation to its azimuth and elevation. This study has focused on enhancing the radar technology used for space debris detection for the purpose of collision avoidance and orbit determination. In 2017, G. Muntoni et al. [12] introduced a bistatic radar configuration using the Sardinia radio telescope for space debris detection and monitoring at 410 MHz in Low Earth Orbit (LEO). The study focused on improving the sensitivity and efficiency of the radio telescope receiver for better space debris monitoring from Earth by detecting the signal scattered by the debris. Currently, not much research has focused on detecting and monitoring LEO space debris using radar sensors onboard small satellites orbiting the Earth.

Radar systems work by the principle of echo (signal sent to an object that reflects it back to the signal source) and this can either be done by one system (called monostatic) or multiple systems (called bistatic or multistatic) [13]. The suitable types of radar for space debris monitoring and tracking are phased arrays, parabolic reflectors, and interferometers [14].

This paper introduces an optimized DNN model for detecting and classifying space debris in real time that can be implemented onboard various payload sensors and other radar devices in small satellites to enhance the detection and monitoring of space debris that are less than 10 cm in size.

2. Methodology

The proposed system can be used on a radar onboard a small satellite that contains the space debris detection model. The radar systems used on ground are limited by their large size, the long distance of detection, and the maximum azimuth and elevation values allowed by the system [13,14]. The proposed model will use DNN following the architecture of target list, classification, data labelling, and filtering. The flow chart of the system can be seen in figure 1 below.

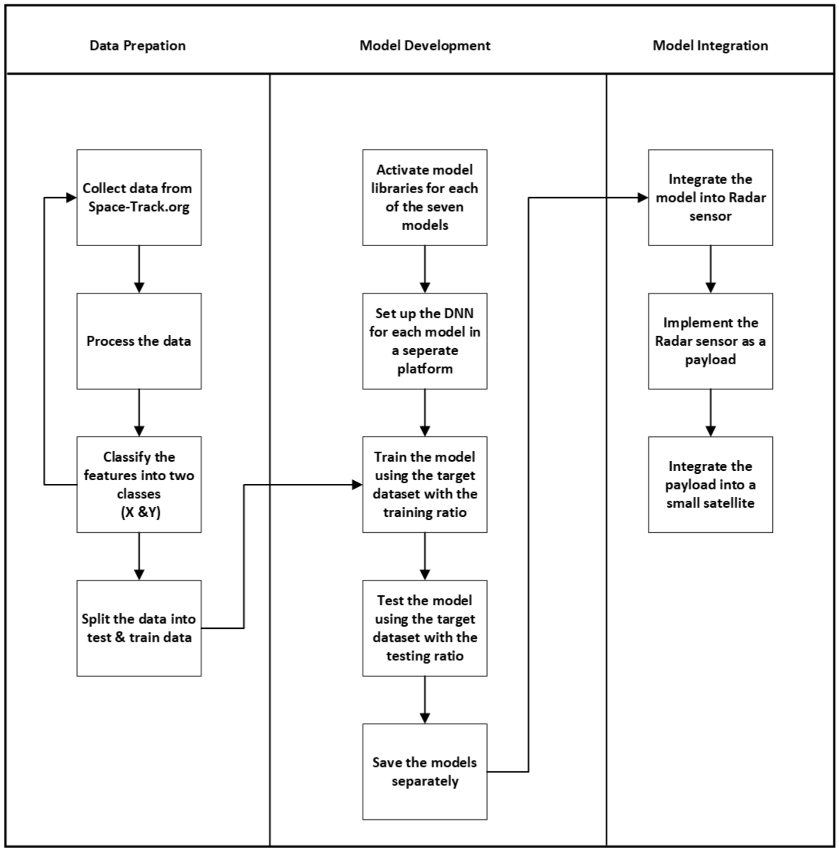

Fig. 1. The proposed system flow chart

2.1 Target list

The target list consists of the space debris objects that the model needs to detect and monitor. These targets are identified based on a predetermined set of criteria such as object size, velocity, altitude, date, time, semi-major axis, radar cross section (RCS), etc. The space object variable in the dataset used is the feature that is needed to be classified. The size of the object detected is obtained to further assist in the classification of the target detected. Historical data was used to train the model to learn the patterns and uncover the relationships between the target object and the other features in the dataset.

2.2 Classification

The classification process is done to identify which category or class an object belongs to through recognition, understanding, and grouping. The architecture of the proposed system has been designed to be simple and scalable to optimize power consumption. It is done using a supervised learning technique to train the model through pt-categorized training datasets to categorize the data into various classes through 'pattern recognition'. By mapping a function (f) from input variables (x) to get the discrete output variables (y). From this, a probability score is generated and assigned to the input to determine the object detected and its size category to classify future data into relevant predetermined categories. The model classifies the detected objects into three size categories based on their size, small, medium, and large. The debris is also classified based on its location, date, time, altitude, and velocity.

2.3 Data labelling

This is part of the data preparation for the deep learning model. This process entails the identification of the raw data and adding one or more labels to that data to specify its setting for the model to enable accurate predictions. This allows the selection of optimal data predictors for the model and typically requires human-in-the-loop participation. This enables the training, fine-tuning, and testing of the model by feeding it with the most applicable datasets for the project. This also results in a trained model that can make predictions on new data. The accuracy of the trained model depends heavily on the accuracy of the data inputted in the system; thus, data labelling is essential to ensure that the developed model is highly accurate.

2.4 Filtering

Filtering is used to identify the relationships between the various parameters in the data. This is done to identify the similarities between the data and the object detected. This process is used to classify the objects into different classes according to the predefined categories which in this case is the size of the object detected.

3. Data preparation

The dataset used was collected from space track application programming interface, as the data were pre-processed as per the requirement to feed the proposed DNN model. Therefore, the database includes various information on currently flying objects in the Earth's orbit including satellites and debris. The current data available for space debris can be seen to verify that the majority of space debris is of small size from 1 cm and below (figure 2).

The aspect of a dataset that is necessary to comprehend more completely is the target variable. The user would wish to forecast this variable using the rest of the dataset. In most cases, the target variable is derived using a supervised ML technique. A feature is a specific quantifiable quality or aspect of a phenomenon. Effective methods for classification and regression depend heavily on the selection of relevant, discriminating, and independent characteristics. In the field of machine learning the variable in a dataset that is required to learn about is called the target variable. A supervised ML algorithm uses historical data to learn patterns and find connections between other parts of the dataset and the target. In this case, after the data was interpreted in the above, the target variables were processed and featured into labels. Each data target had a sample size corresponding to the other, as all null samples were neglected. The main variables used in this study are seen in Table 1.

Table 1. Major data for space debris detection from dataset.

Data no.	Variable	Data type	Mean
1	Object age	Object	-
2	Object altitude	Object	Earth/LEO
3	Apoapsis	Object	5721.1
4	Periapsis	Object	2795
5	Mean_anomaly	Float 64	191.2
6	Mean_motion	Float 64	12.5
7	Inclination	Float 64	74.4
8	Period	Float 64	223.5
9	Debris size	Object	1*

* 1: small, 2: medium, 3: large

After the target variables shown in Table 1 were processed and cleaned, they were then classified into two feature sets (X and Y), the X feature had the target data of the debris object age, object altitude, apoapsis, periapsis, mean anomaly, mean motion, inclination, and period. While, the Y feature, contained, debris size collected historically from the RCS.

4. Model development

Six models (Bernoulli Naïve Bayes, Gaussian Naive Bayes, Logistic Regression, Support Vector Machine, K-Nearest Neighbour, and Random Forest Classifier) were developed with respect to the concept of machine learning and deep learning neural network, in addition, one booster model (AdaBoost Classifier) all in aim to utilize from the boosting principle where it has been a common way to solve problems with two possible answers. By turning some weak learners into strong learners, these algorithms improve the ability to predict furthermore. The models were developed after the concept of deep learning and machine learning. Since, deep learning algorithms can improve their results through repetition without human interaction, while machine learning algorithms typically require human correction when error occurs. A deep learning algorithm needs vast datasets, which may include heterogeneous and unstructured material, in contrast to a machine learning algorithm, which can learn from relatively small amounts of data. ML is considered evolving into deep learning. Hence, a machine learning method known as deep learning layers algorithms and computational units, or neurons, develop into a structure known as a deep neural network. These DNN are modelled after how the human brain is formed, as this network of interconnected algorithms processes data in a non-linear way. Therefore, each of the seven models were developed with respect to their working principle. The model training requires a powerful computer with a powerful processor and graphics card. The model was trained using Intel(R) Core (TM) i7-9750H CPU @ 2.60GHz 2.59 GHz installed RAM 32.0.

4.1 Bernoulli Naive Bayes

The Bernoulli Naive Bayes (BNB) model [15] was developed since it is used in ML algorithms when the dataset has a binary distribution, and the output label is either present or absent. Therefore, the development of this model is used effectively. However, the algorithm's primary benefit is that it accepts features as binary values. After the data was prepared it went through the algorithm in the manner shown in Table 2.

Table 2. The input used to develop the BNB algorithm.

Model	Train	Test	Random state
Bernoulli Naive Bayes	70	30	0

4.2 Gaussian Naive Bayes

A classification method used in ML called Gaussian Naive Bayes (GNB) algorithm [16] is based on the probabilistic method and Gaussian distribution. Each parameter that is identified as a feature or a predictor is presumed to have an independent capacity to predict the output variable via GNB. Hence, the dataset underwent the algorithm with the manner shown in Table 3.

Table 3. The input used to develop the GNB algorithm.

Model	Train	Test	Random state
Gaussian Naive Bayes	70	30	0

4.3 Logistic Regression

A predictive analytical algorithm based on the idea of probability; logistic regression [17] is an ML approach that is used for classification challenges. The maximum iteration property shows how many times a formula cell can be evaluated before it is marked as unresolved. By setting the maximum iteration property, cells can be kept from being evaluated an infinite number of times when they have circular references. The maximum number of iterations taken for the solvers to converge value set was 500000. So, the dataset underwent the algorithm in the manner shown in Table 4.

Table 4. The input used to develop the logistic regression model.

Model	Max iteration	Train	Test	Random state
Logistic Regression	500000	70	30	49

4.4 Support Vector Machine

A deep learning system known as a support vector machine (SVM) [18] uses supervised learning to classify or predict the behaviour of groupings of data. Supervised learning systems in AI and ML give input and intended output data that are labelled for classification. The SVM working principle is basically, finding a hyperplane in N-dimensional space (N is the number of features) that categorizes the data points clearly which is the goal of the SVM algorithm. There are a variety of different hyperplanes that might be used to split the two classes of data points, so finding a plane with the greatest margin that is the greatest separation between data points from both classes is the goal. Maximizing the margin distance adds some support and increases the confidence with which future data points can be classified. So, the dataset underwent the algorithm in the manner shown in Table 5.

Table 5. The input used to develop the SVM.

Model	Estimator	Train	Test	Random state
Support Vector Machine	10	75	25	0

4.5 K-Nearest Neighbour

The k-nearest neighbours' (KNN) algorithm [19] is a supervised learning classifier that uses proximity to produce classifications or predictions about how a particular data point will be grouped. It is a non-parametric classifier. Therefore, the supervised learning technique KNN is used for both regression and classification. By calculating the distance between the test data and all the training points, KNN tries to predict the proper class for the test data. Then chooses the K sites that are closest to the test data. The KNN method determines which classes of the "K" training data the test data will belong to, and the class with the highest probability is chosen. The value in a regression situation is the average of the 'K' chosen training points. The model development focused on Minkowski metric to train and test the model with since the Minkowski metric is a metric for vector spaces with real values. Minkowski distance can only be calculated in a normed vector space, which is a space where distances can be shown as vectors with lengths that cannot be negative. Minkowski distance is in a generalized form, and we can change it to get different distance metrics when changing the 'p'. The P value was set as 2. Therefore, mathematically, having the value of p = 2, we will get the Euclidean distance between two points in Euclidean space which is the length of a line segment between the two points. So, the dataset underwent the algorithm in the manner shown in Table 6.

Table 6. The input used to develop the KNN

Model	Metric	Train	Test	Random state
K-Nearest Neighbour	Minkowski Point = 2	80	20	0

4.6 Random Forest Classifier

As its name suggests, a random forest (RFC) [20] is made up of numerous independent decision trees that work together as an ensemble. The class with the highest votes becomes the model's prediction, which is generated by each individual tree in the random forest. Therefore, the ML algorithm random forest is a part of the supervised learning methodology. It can be applied to ML issues involving both classification and regression. It is built on the idea of ensemble learning, which is a method of integrating various classifiers to address difficult issues and enhance model performance. Some decision trees may predict the correct output, while others may not, because the random forest combines numerous trees to forecast the class of the dataset. But when all the trees are combined, they forecast the right result. Two assumptions were made for the random forest, the first assumption is for the dataset's feature variable to predict true outcomes rather than a speculated result, as there should be some actual values in the dataset. The second assumption is there must be an extremely low correlation between each tree's predictions. So, the dataset underwent the algorithm in the manner shown in Table 7.

Table 7. The input used to develop the RFC.

Model	Estimator	Depth	Random state
Random Forest Classifier	100	10	1

4.8 AdaBoost Classifier

The abbreviation "AdaBoost" stands for "Adaptive Boosting" [21], which is a general approach used to turn several "poor classifiers" into one strong classifier. An AdaBoost classifier is a meta-estimator that first fits a classifier on the original dataset, and then it fits additional copies of the classifier on the same dataset with the weights of instances that were incorrectly classified being changed so that later classifiers would concentrate more on challenging cases. So, the dataset underwent the algorithm in the manner shown in Table 8.

Table 8. The input used to develop the AdaBoost Classifier.

Model	Train	Test	Random state
AdaBoost Classifier	80	20	0

5. Results

The harmonic mean of precision and recall is provided by the f1-score. The fl-score is the score assigned to each class that will indicate how accurately the classifier classified the data points inside that class in comparison to all other classes. The number of samples of the real response that belong to that class serves as the support. The tables below show the results obtained from the system.

Table 9. Summary of all the algorithmic models

Model	Accuracy Score	Accuracy Percentage
Bernoulli Naive Bayes Algorithm	0.900	90%
Gaussian Naive Bayes Algorithm	0.917	~ 92%
Logistic Regression	0.926	~ 93%
Support Vector Machine	0.928	~ 93%
K-Nearest Neighbour	0.929	~ 93%
Random Forest Classifier	0.936	~ 94%
AdaBoost Classifier	0.950	95%

Table 10. BNB Algorithm

Run	Precision	Recall	F1-Score	Support
1	0.97	0.95	0.96	1471
2	0.47	0.78	0.58	133
3	0.00	0.00	0.00	53
Accuracy			0.90	
Macro average	0.48	0.58	0.51	1657
Weighted average	0.90	0.90	0.90	1657

Table 11. GNB Algorithm

Run	Precision	Recall	F1-Score	Support
1	0.97	0.96	0.96	1471
2	0.59	0.67	0.63	133
3	0.43	0.40	0.41	53
Accuracy			0.92	
Macro average	0.66	0.67	0.67	1657
Weighted average	0.92	0.92	0.92	1657

Table 12. Logistic Regression

Run	Precision	Recall	F1-Score	Support
1	0.95	0.98	0.97	2248
2	0.62	0.49	0.55	164
3	0.85	0.23	0.36	74
Accuracy			0.93	2486
Macro average	0.81	0.57	0.62	2486
Weighted average	0.92	0.93	0.92	2486

Table 13. SVM

Run	Precision	Recall	F1-Score	Support
1	0.95	0.99	0.97	1843
2	0.65	0.61	0.63	161
3	0.00	0.00	0.00	67
Accuracy			0.93	
Macro average	0.53	0.53	0.53	2071
Weighted average	0.89	0.93	0.91	2071

Table 14. KNN

Run	Precision	Recall	F1-Score	Support
1	0.95	0.98	0.97	1471

2	0.66	0.59	0.62	133
3	0.89	0.32	0.47	53
Accuracy			0.93	
Macro average	0.84	0.63	0.69	1657
Weighted average	0.93	0.93	0.92	1657

Table 15. AdaBoost Classifier

Run	Precision	Recall	F1-Score	Support
1	1.00	0.89	0.94	9
2	0.92	1.00	0.96	11
Accuracy			0.95	20
Macro average	0.96	0.94	0.95	20
Weighted average	0.95	0.95	0.95	20

5.1 KNN Validation

After the model was developed, trained, and tested, a graph was generated to show the correlation between the 'X' train split which is labelled as 0, the 'Y' train split, labelled as 1, the 'X' test split labelled as 0, and the 'Y' test split labelled as 1. The validation can be seen in figure 2 below.

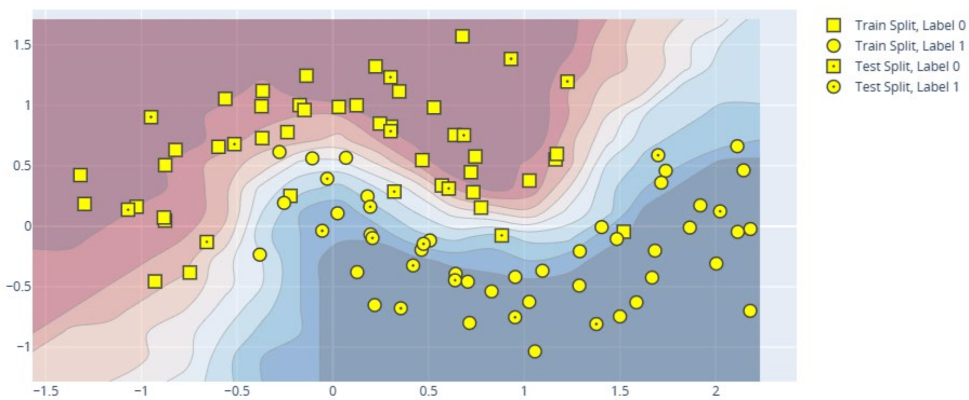

Fig. 2. KNN validation

5.2 Random Forest classifier validation

The data target variables from the dataset were utilized as the input for the random forest model. Hence, the report of the random forest classifier is generated in a decision tree shown in figure 3. The X target variables were grouped into two classes of [0,1].

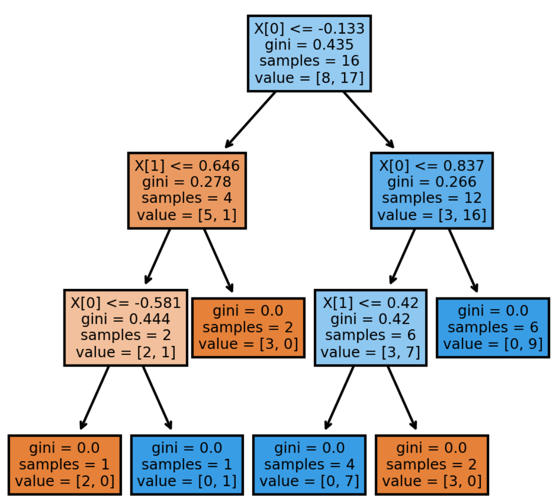

Fig. 3. Random Forest decision tree

6. Discussion

Space debris is considered the number one risk in space missions, as well as its direct effect on achieving the sustainable development goals (SDGs). Space debris ranging from sizes 1 mm to 10 cm requires enhanced tracking and monitoring methods to effectively monitor and track space debris. Therefore, utilizing payloads onboard small satellites with sensors/radars technology that is integrated with DNN and ML models, is an option to a solution to detect, track and monitor space debris. Seven models were developed with respect to the DNN and ML techniques and working principles. The proposed model contained DNN architecture of target list, classification, data labelling, and filtering. The target data were feature classified into two classes, with respect to the data labelling and filtering. All in aim to promote high accuracy scores when learning from the historical RCS data.

Each of the seven models were trained and tested separately, with respect to the algorithm parameters, including the ratio of the training and testing of the processed target dataset. The results obtained from the seven models, demonstrated above, showed that the highest accuracy score between the models was the RFC of around 94%, while the lowest was BNB of around 90%, However, using the booster the accuracy score was increased by 1% to about 95%. The overall average accuracy was found to be 92.86%.

The proposed system's accuracy can be further enhanced by using a 4-dimensional (4D) radar system. This system is inspired by the proposed K-Radar system by Paek et al [22] using the 4D radar tensor-based 3D object detection dataset and benchmark from KAIST-Radar. The proposed radar system to be developed for onboard the satellite will contain the needed parameters discussed above in addition to the power measurement with the doppler, range, azimuth, and elevation dimensions. The system will also include a stereo camera to cover the front field of view (FOV) of the spacecraft with a Global Positioning System (GPS) to enable accurate positioning of the spacecraft and the detected debris with respect to the satellite and obtain the 4D view of the surroundings. The system can provide a unique tracking ID for each annotated object to be used to track the surrounding space debris along a sequence of frames obtained from the camera and radar along with the seven detection models onboard that are trained to detect small-sized space debris. This radar system with the developed proposed detection model will maximize the efficiency of the overall detection, monitoring, and tracking system for space debris.

7. Conclusions

Space debris is a major concern to all space-faring countries and many studies and research have been conducted on the detection and monitoring of the debris, however, there is still a gap in the tracking and monitoring of small-sized debris of less than 10 cm in size. This study proposed an optimized space debris detection and classification system using a DNN model that can be used onboard of small satellites such as CubeSats for real-time monitoring and detection through radars or sensors. The proposed system works by developing the DNN model that can be integrated into any radar system or sensor to detect small space debris. Additionally, the system was shown to detect and classify objects with an accuracy of range from 90% to 94%, while by using a booter it reached 95%, and the overall average accuracy score was found to be around 93%. This has shown that it is applicable to be implemented for real-time tracking of space debris.

In the future, the system can be investigated further by studying its feasibility onboard a spacecraft such as a CubeSat and integrating it with a radar system to demonstrate this technology and test the system in space to evaluate its performance and classification accuracy in orbit.

Acknowledgments

The authors acknowledge Space Track and the National Space Science Agency for the valuable resources that supported this paper.

References

[1] S. Hobbs et al., "Space waste," Waste, pp. 567–583, Jan. 2019.

[2] M. Garcia, "Space debris and human spacecraft," NASA, 14-Apr-2015. [Online]. Available: https://www.nasa.gov/mission_pages/station/news/orbital_debris.html.

[3] M. Zhu, "A break-even analysis of orbital debris and space preservation through monetization", Journal of Space Safety Engineering, 2022. Available: https://www.sciencedirect.com/science/article/abs/pii/S2468896722001045.

[4] J. Peltoniemi et al., "Light scattering model for small space debris particles", Advances in Space Research, 2022. Available: 10.1016/j.asr.2022.09.022.

[5] R. Wickramatunga, "United Nations Office for Outer Space Affairs," ST/SPACE/71 - European Global Navigation Satellite System and Copernicus: Supporting the Sustainable Development Goals. Building Blocks towards the 2030 Agenda", 2018. [Online]. Available: https://www.unoosa.org/oosa/en/oosadoc/data/documents/2018/stspace/stspace71_0.Html.

[6] A. Bouquet et al., "Simulation of the capabilities of an orbiter for monitoring the entry of interplanetary matter into the terrestrial atmosphere," Planetary and Space Science, vol. 103, pp. 238–249, 2014.

[7] E. Kim et al., "Shomy: Detection of small hazardous objects using the You Only Look Once algorithm," KSII Transactions on Internet and Information Systems (TIIS), vol. 16, no. 8, pp. 2688–2703, 2022.

[8] R. Ravindran, M. Santora and M. Jamali, "Multi-Object Detection and Tracking, Based on DNN, for Autonomous Vehicles: A Review", IEEE Sensors Journal, vol. 21, no. 5, pp. 5668-5677, 2021. Available: https://ieeexplore.ieee.org/abstract/document/9274366.

[9] Z. Zhao, P. Zheng, S. Xu and X. Wu, "Object Detection With Deep Learning: A Review", IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212-3232, 2019. Available: https://ieeexplore.ieee.org/abstract/document/8627998.

[10] C. Szegedy, A. Toshev and D. Erhan, "Deep Neural Networks for Object Detection", Advances in Neural Information Processing Systems, pp. 2553-2561, 2013. Available: https://proceedings.neurips.cc/paper/2013/hash/f7cade80b7cc92b991cf4d2806d6bd78-Abstract.html.

[11] A. Morselli et al., "A new high sensitivity radar sensor for space debris detection and accurate orbit determination", IEEE, 2015.

[12] G. Muntoni et al., "Space Debris Detection in Low Earth Orbit with the Sardinia Radio Telescope", Electronics, vol. 6, no. 3, p. 59, 2017. Available: 10.3390/electronics6030059.

[13] M. I. Skolnik, Radar Handbook, 2nd ed. Boston, MA: McGraw Hill, 1990.

[14] G. Muntoni, G. Montisci, T. Pisanu, P. Andronico, and G. Valente, "Crowded space: A review on radar measurements for space debris monitoring and tracking," Applied Sciences, vol. 11, no. 4, p. 1364, 2021.

[15] M. Artur, "Review the performance of the Bernoulli Naïve Bayes Classifier in Intrusion Detection Systems using Recursive Feature Elimination with Cross-validated selection of the best number of features," Procedia Computer Science, vol. 190. Elsevier BV, pp. 564–570, 2021. doi: 10.1016/j.procs.2021.06.066.

[16] D. Petschke and T. E. M. Staab, "A supervised machine learning approach using naive Gaussian Bayes classification for shape-sensitive detector pulse discrimination in positron annihilation lifetime spectroscopy (PALS)," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 947. Elsevier BV, p. 162742, Dec. 2019. doi: 10.1016/j.nima.2019.162742.

[17] N. Wichitaksorn, Y. Kang, and F. Zhang, "Random feature selection using random subspace logistic regression," Expert Systems with Applications, vol. 217. Elsevier BV, p. 119535, May 2023. doi: 10.1016/j.eswa.2023.119535.

[18] H. Moosaei, M. A. Ganaie, M. Hladík, and M. Tanveer, "Inverse free reduced universum twin support vector machine for imbalanced data classification," Neural Networks, vol. 157. Elsevier BV, pp. 125–135, Jan. 2023. doi: 10.1016/j.neunet.2022.10.003.

[19] Z. Xu, J. Cao, G. Zhang, X. Chen, and Y. Wu, "Active learning accelerated Monte-Carlo simulation based on the modified K-nearest neighbors algorithm and its application to reliability estimations," Defence Technology. Elsevier BV, Sep. 2022. doi: 10.1016/j.dt.2022.09.012.

[20] T. Bergmayr, S. Höll, C. Kralovec, and M. Schagerl, "Local residual random forest classifier for strain-based damage detection and localization in aerospace sandwich structures," Composite Structures, vol. 304. Elsevier BV, p. 116331, Jan. 2023. doi: 10.1016/j.compstruct.2022.116331.

[21] G. Hu, C. Yin, M. Wan, Y. Zhang, and Y. Fang, "Recognition of diseased Pinus trees in UAV images using deep learning and AdaBoost classifier," Biosystems Engineering, vol. 194. Elsevier BV, pp. 138–151, Jun. 2020. doi: 10.1016/j.biosystemseng.2020.03.021.

[22] D.-H. Paek, S.-H. Kong, and K. T. Wijaya, "K-RADAR: 4D radar object detection for autonomous driving in various weather conditions," OpenReview, 06-Jun-2022. [Online]. Available: https://openreview.net/forum?id=W_bsDmzwaZ7.